Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/5540--11-9350

Determining Asset Criticality for Cyber Defense

Anya Kim
Myong H. Kang

Center for High Assurance Computer Systems
Information Technology Division

September 23, 2011

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 23-09-2011	**2. REPORT TYPE** Memorandum Report	**3. DATES COVERED** *(From - To)* September 2011–May 2011

4. TITLE AND SUBTITLE Determining Asset Criticality for Cyber Defense	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 0603235N
6. AUTHOR(S) Anya Kim and Myong H. Kang	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER 55-6334
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory, Code 5540 4555 Overlook Avenue, SW Washington, DC 20375-5320	**8. PERFORMING ORGANIZATION REPORT NUMBER** NRL/MR/5540--11-9350
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research One Liberty Center 875 N. Randolph St., Suite 1425 Arlington, VA 22203-1995	**10. SPONSOR / MONITOR'S ACRONYM(S)** ONR
	11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Current cyber network defense practices lack a standard methodology to properly determine event priority. Events are generally handled on a first-come first-serve basis. Some limited knowledge of target assets is applied, but in a non-standard manner based on the decision-maker's domain-specific knowledge. This not only requires proficient domain expertise, but is also very manpower intensive. We need an asset criticality metric that enables users to address events that target critical assets first. Determining asset criticality is not a trivial problem. The various contributing factors must be identified and combined. Hierarchical missions and commands that they support must be considered. Dependency relationships should also be factored in. In this paper, we report our ongoing research for determining asset criticality.

15. SUBJECT TERMS

Asset criticality Decision making

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Anya Kim
a. REPORT Unclassified Unlimited	**b. ABSTRACT** Unclassified Unlimited	**c. THIS PAGE** Unclassified Unlimited	UU	35	**19b. TELEPHONE NUMBER** *(include area code)* (202) 767-6698

Determining Asset Criticality for Cyber Defense

1. Introduction

Consider two missiles targeting two cities simultaneously. You are responsible for the safety of these cities but have the time and resources to defend only one. Which city would you defend? With no additional information, deciding which missile to respond to becomes a difficult problem. Knowledge about the missiles – e.g., one has a bigger payload – may help you decide. But information about the missiles alone is not sufficient to understand the consequences of your decision. Information about the target cities, such as one is a densely populated metropolis while the other is a sparsely populated desert area – may be required to understand the impact of the decision. This may tell you that the effects of the smaller missile targeting the largely populated city would lead to devastating catastrophic results on the people and wildlife with the effects lasting for many years, while the larger missile would lead to minimal loss of lives in the desert. While every life is precious, the decision here may seem obvious. Then again, further information about the target cities such as the G-20 summit is being held at the sparsely populated city may result in the importance of the cities being reversed. Regardless of the choice you ultimately make, it is obvious that the more information you have about the cities, the better off you will be because you will make more informed decisions.

Such is the argument for asset criticality in the cyber domain where 'missiles' are analogous to cyber attacks, and the 'cities' are the hosts/assets that are potential targets for cyber attacks. In cyber defense, there are various tools that can tell us about the 'missiles' – e.g., the trajectory and strength, but little is known about the relative importance of the 'target cities'. Some tools can provide pieces of information about the assets. However, there is no standard way to combine this information to get a "standardized" asset criticality (AC) metric that indicates the relative importance of assets. Information about the assets – i.e. factors – needs to be combined with cyber event data to get a true understanding of the priority of the incident and to determine proper course of actions. Figure 1 shows how factors are combined to obtain an asset criticality metric, and how this value can be used to prioritize events. The decision-making process takes into consideration asset criticality, event priority, state of the asset (e.g., patches up-to-date) and target space to recommend a course of action that is based on the whole picture.

Manuscript approved June 21, 2011.

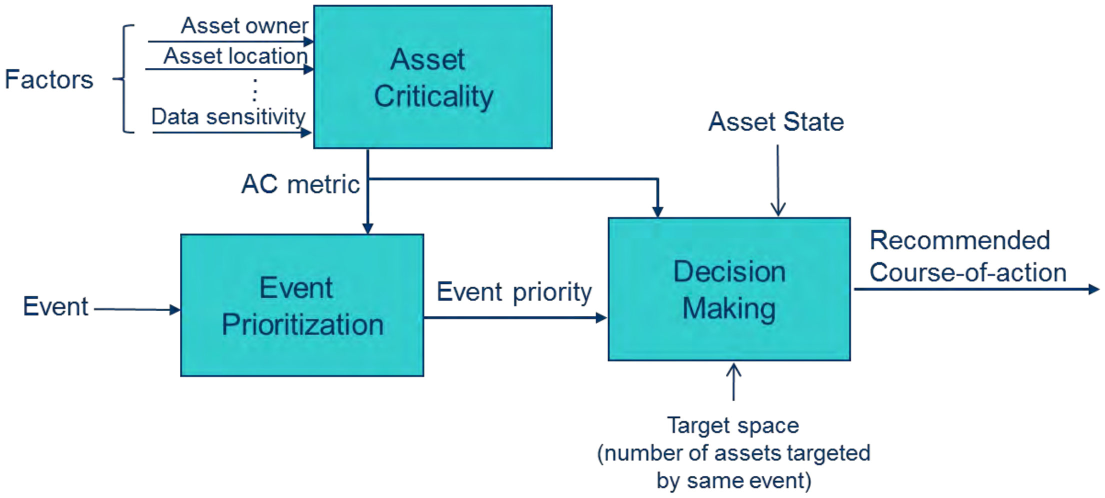


Figure 1. An Event Prioritization Process

The military (or any organization for that matter) has many assets that provide mission support. Missions are composed of other missions. Some missions are more important than others. Also some assets are more critical to a mission than others. Which assets are more critical and to what degree depends on many factors such as the definition of criticality in terms of mission support, how an asset contributes to mission success, and the nature of the mission-related applications running on the asset. These are all potential candidates to aid in determining asset criticality. These factors determine the criticality of an asset in different ways: some may change the definition of what it means to be critical, some contribute directly, others indirectly, some factors may have values that change frequently (e.g., login information), providing a dynamic and challenging environment in which to determine asset criticality. The objective of this paper is to take these factors and combine them in a way that enables scoring and ranking of assets based on their criticality to the overall mission. Cyber warriors can use these scores and rankings to efficiently prioritize their responses based on both the severity of the incoming attacks and the criticality of the target assets.

The rest of the paper is organized as follows. Section 2 describes the environment for which we are providing asset criticality metrics[1] and the conditions/limitations that must be met within the scope of that environment. A high level overview of our approach is provided in section 3. Section 4 examines potential methods that can be applied to develop an asset criticality metric. Section 5 describes our approach in detail, applying methods from section 4 as relevant. A detailed example of our method is provided in section 6. We conclude the paper in section 7.

2. Background

The current operational process for handling cyber threats within the DoD involves a handful of personnel dealing with a large number of cyber events spread across hundreds of locations including ships, medical clinics, headquarters, and research laboratories [**1**]. These cyber warriors assume different roles (e.g., watch stander, incident handler) using a suite of security tools to monitor, report and thwart malicious network activity. The tools provide a

[1] Throughout this paper we use the terms AC metric, AC score and AC value interchangeably.

comprehensive and holistic view of security events so that the monitoring team can immediately act on the most critical events [**2**]. While the tools and expertise of the team enable these cyber warriors to obtain information about the events (such as attack paths, likelihood of successful compromise, the nature and severity of the event, etc.), information about the potential risk to an asset is not readily available using any tools or software. Some prioritization including assets is performed, but uses the human decision-maker's internal knowledge that is different for each decision-maker. This requires proficient domain knowledge and substantial time. With a handful of cyber warriors defending the DoD's computer networks against hundreds of thousands of network events on a daily basis, determining which events to handle first, and what proper course of action should be taken can be a challenging task. The cyber warrior needs an autonomous way to determine course of actions. The event prioritization process should combine the severity of the events with the criticality of the asset that is being targeted. A crucial piece of the puzzle that is missing for autonomous decision making and event prioritization is an asset criticality metric that provides information about the criticality of an asset. For example, if two similar events are detected, targeting two different host machines, the cyber warrior needs to determine which event to address first. If he is able to determine that one target is a weapons system and the other is a generic desktop, proper course of actions can be easily prioritized and determined. This will ultimately lead to a more precise, tailored, and faster response.

The need for an asset criticality metric is particularly strong in the DoD as opposed to commercial networks because the military must constantly protect itself against structured and advanced persistent threats. These threats are directly targeted at DoD leadership, infrastructure and/or processes. Unlike the commercial sector, loss of critical assets directly leads to loss of lives.

3. Basic Approach

While determining asset criticality is an important problem, particularly for the DoD, it is far from trivial. The DoD is composed of four military branches (Army, Navy, Air Force, Marine Corps) and various non-combat agencies (such as NSA, DIA, etc.) each with its own military mission and organizational structure to accomplish that mission. Within each service, various commands conduct operations related to logistics, intelligence gathering, operational combat, infrastructure support, etc. These missions support and complement each other to ultimately achieve the higher-level DoD mission [**3**]. For these purposes, the command structure of the DoD is hierarchically organized with units, battalions, divisions and brigades composing smaller units that work jointly.

The structure of the DoD computing environment is similar to that of the DoD itself – each command has computer assets that perform various duties. The duties range from mundane tasks, such as web hosting, to highly critical tasks, such as weapons systems, to support the various missions in varying degrees. These assets are commonly known as the Global Information Grid (GIG) [**4**]. For cyber network defense, sensor data collected from myriad sources are brought up to the higher level system that aggregates, correlates and processes the data to present an integrated picture. In short, the organization, chain-of-command, etc. in the DoD are structured to ensure successful mission completion and continuation of operations. Since the DoD is a very mission-oriented organization, asset criticality must be defined in terms of how crucial the asset is to the success of the mission, where missions themselves can be composed of hierarchies.

Obviously, an asset running or taking part in a critical mission is itself a critical asset. Unfortunately, an asset's contribution level to a mission is not directly observable. Thus, asset criticality needs to be estimated from other attributes of the assets that we call *factors* or *criteria*. For example, the purpose of a host machine (e.g., is it a personal desktop computer, or a database server), the operating system it is running (e.g., mission-critical software generally runs on Linux-based machines), or the types of mission-level applications running on the asset are factors that can be used to determine/estimate the criticality level of an asset. The fact that an asset is located within a specific command, or is located in a certain geographic region may also contribute to its criticality. For example, two identical assets, one located in a peaceful area and one in a wartime region may differ in mission support. Furthermore, in the military, the fact that a particular person is using the asset immediately elevates the criticality level of the asset despite what it's being used for. This is analogous to the Air Force One call sign. Typically, the term refers to those aircraft whose primary mission is to transport the President of the United States – however, any (U.S. Air Force) aircraft carrying the President can use the call sign of Air Force One while he is on board [**5**].

Factors also have differing degrees of importance when measuring asset criticality. For example, while the type of applications running on an asset and the type of operating system being used may both contribute to the criticality of the asset, one may contribute more to the criticality of the asset than the other. This degree of importance also needs to be determined. The degree of importance (i.e. level of contribution) of each factor is called its weight.

Identifying the factors that contribute to the criticality of an asset and determining how to combine them in a meaningful way are the first step in determining asset criticality. While being able to rank the assets from most critical to least critical is useful, we need to have a score for each asset in order to determine the level of criticality for each asset. Whatever approach we use, the calculated scores need to reflect the relative criticality of the asset (i.e. its importance to the overall mission) instead of just providing a ranking of the assets.

For our purposes, we have categorized the factors into external, static, and value-sensitive factors based on the way they affect different aspects of criticality and how to measure them. *External factors* are those that may determine the definition of criticality and the underlying criteria used to judge the criticality of an asset. For example, triage of wounded soldiers is handled differently between wartime and peacetime [**6**]. In times of peace, the most critical soldiers are given priority treatment first, similar to civilian triage techniques. However, in wartime, lightly wounded soldiers receive priority treatment, so they can be quickly sent back out to the battlefield [**7**]. This concept can apply to asset criticality; a tactical asset may have priority over a strategic asset in some cases, and vice versa in other situations. External factors are those that reflect these situations.

Static factors are factors with values that do not change over long periods of time and can be used to indirectly assess the criticality of an asset with respect to its mission. Examples of static assets are the types of application on the asset, its purpose, underlying operating system, etc. These factors can be combined and pre-calculated to obtain initial AC metric values that can be adjusted later by applying external and value-sensitive factors.

Value-sensitive factors are those that change the criticality of an asset only when they have a certain value. When value-sensitive factors possess this specific value, they raise the criticality of an asset regardless of what other values in other factors the asset possesses. For example, who is currently logged onto the machine is a factor that changes the criticality of an asset depending on the value. In most instances, who is logged on may not matter to the criticality level of the asset. However, if a highly ranked person such as an admiral or the President of the United States is using the machine then the criticality of the machine is elevated to the utmost importance regardless of all other factors. In other words, even if the machine is not being used for mission-critical tasks, the high rank of the user (i.e. his role) makes the asset rank highly on the critical list. These value-sensitive factors can be further categorized as dynamic value-sensitive and static value-sensitive. Who's logged on is an example of a dynamic value-sensitive factor with frequently changing values. Who owns the machine is an example of a static value-sensitive factor whose value remains relatively constant. This should not be confused with static factors described above. Value-sensitive factors are categorized this way because they are handled differently. Static value-sensitive factors can be calculated at the earliest stage. Assets that possess specific values for these factors immediately obtain a high AC metric value and further processing of additional factors need not be performed. Dynamic value-sensitive factors need to be assessed at run-time, because the value of the factors may change and therefore have to be reassessed every time the criticality of an asset is evaluated.

One may think that determining asset criticality is a trivial problem: just take the factors, multiply them by their weights and add them up to obtain a score. However, examining these different factors shows us that they need to be processed in different manners. Furthermore, with several hundred thousands of assets spread across land, sea, and ships, we cannot expect one or two people to possess all the domain knowledge required to rate these assets. We cannot even expect to use a standard set of factors to evaluate asset criticality. For example, a research lab's primary mission may be developing tools and techniques for strategic warfare, while the U.S. Fifth Fleet's primary mission may be engaging in tactical warfare. Therefore, the importance of assets in a research lab may be evaluated differently than that in the 5th Fleet unit. Each command needs to rate its assets based on the importance to their own mission.

Ultimately, people (or systems) that are most knowledgeable about a set of assets will need to score and rank these assets. That means that different entities will be assessing different groups of assets. This implies that while decision-making and asset criticality assessment may be an autonomous system, the criteria/factors used to judge the assets, the importance of the criteria (i.e. weights), and the decision-making process employed may differ from command to command.

When different users assess different sets of assets, and thus subjective evaluations are involved, comparing two different assets' AC scores from two different commands may not be accurate. For example, the decision maker in each command may feel that all his assets are critical and give them all high values. Therefore, after each asset is rated individually, the importance of the command they belong to must also be incorporated into the measurement to obtain a more accurate representation of the criticality of an asset with respect to the entire organization. The importance of a command is not a constant value. External factors can change how important commands are in relation to one another. Taking the 5th Fleet example from earlier, the missions themselves can change depending on whether it is wartime or peace time (i.e. external factors). For example, the 5th Fleet in wartime primarily engages in tactical combat, while during peace time it may protect commercial ships from pirates. This in turn may make the 5th Fleet more mission critical during wartime than peace time, with the effect trickling down to its assets. As in the case of measuring asset criticality, there is no one entity that can possess knowledge of all the other commands. Commands report up to other commands, following the chain-of-command structure. Hence, a commander has knowledge of its own mission, as well as the mission of the commands immediately under it.

While the military structure is very hierarchical in nature, there are some cross-communications. Similar to the way that various organizations within the military form Communities of Interest (COI), various assets can have dependency relationships in which assets depend on input from other systems. These systems may belong to the same command, or they may span across different commands. Figure 2 depicts a hypothetical description of a hierarchical command structure with each command possessing its own set of assets (represented as squares). The colored assets represent groups of dependencies in which one or more asset may rely on input from other assets with the same color. These 'islands' of dependencies may make a non-critical asset highly critical if several critical assets depend on it for input: if a dependent host is 100% dependent on input from another host, and cannot complete its mission without the information from this other host, then the provider machine should be as critical as the dependent machine. Therefore, we need to examine dependency relationships so that criticality based on dependency can also be captured.

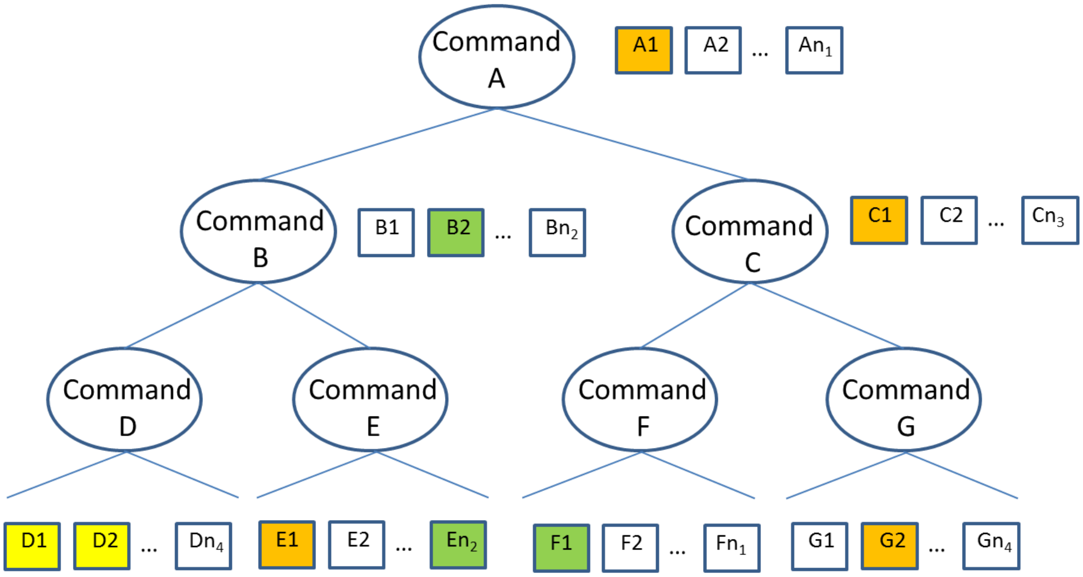


Figure 2. Hypothetical rendering of islands of assets and islands of dependencies

In summary, asset criticality cannot be measured in one step by one entity. Groups of assets sharing a common mission must be measured together using factors and weights that are significant to the specific mission of that group. However, to make these individual measurements scale and be proportional across commands, the importance of the commands also need to be taken into account. Furthermore, there must be a way to capture dependency relationships.

We are trying to model islands of assets where islands are arranged hierarchically, with islands of dependencies interspersed throughout. There is no one model that can be taken from the literature to fit our model and incorporate all its characteristics to obtain asset criticality measurements. Therefore, we take a divide-and-conquer approach where assets are rated in several steps using different methods, and the results are glued together in a cohesive, compatible, and comparable manner.

The method to calculate the asset criticality metric should also meet the following criteria:

- The criticality scores of the assets should reflect the relative criticality of the assets: The criticality rank of an asset is useful for determining, for example, which 100 assets are the most critical. However, the rankings alone do not provide sufficient information. The AC metric should not be just a ranking of the assets but a normalized value that tells us how critical an asset is with respect to other assets. With this AC metric we can determine how much more critical an asset is over another.
- The criticality scores need to be compatible with each other despite the fact that they are calculated by separate commands: Even if different criteria and weights are used to assess different groups of assets, the normalized results should provide a consistent measurement and relative scoring of assets within the same command. This enables asset criticality scores from different commands to be comparable to each other after command-level decisions have been applied.
- The criticality scores should provide near real-time (NRT) calculations: The algorithm should minimize calculation time to provide near-real time posturing of asset criticality. In other words, we should not have to recalculate the scores of the entire set of assets when there are changes in the criticality scores of a few assets due to changes in the values of their factors.

The following is an overview of our algorithm to score and rank assets considering our model of hierarchical islands of assets and the islands of dependencies, while satisfying the constraints mentioned above.

1. Rate each asset individually for each group/command
    a. Use the information available from each command to determine factors and weights that best measures the criticality of its assets in relation to its own missions.
    b. While it is best to use the same criteria and weights across all the commands, this may not always be possible. However, the final measurements need to be compatible across different commands. By compatibility we mean that the units of measurements are the same, even if different factors and weights are used.
2. Rate the commands
    a. Different external situations may affect the importance of different commands. We need to rate commands according to their level of importance, but still make sure that lower level commands are not unfairly penalized by a trickle-down effect.
    b. No one person has information about all the commands. Each upper level command is expected to know its own mission and the mission of those commands directly under it.

   c. The command-level values obtained here are used as weights to level out the importance of different assets in different commands. This enables assets from different commands to be comparable across the commands.

3. Adjust the criticality scores based on the islands of dependencies
   a. Factor in the islands of dependencies to allow input-related dependencies to be captured and reflected in the AC metric.
   b. The adjusted asset criticality score should be a function of the initial AC metric of the asset, obtained from the previous steps, the dependent assets' scores, the weight of the dependencies and the number of dependent assets.
4. Rate the assets based on value-sensitive factors for those assets that possess critical values in these factors
   a. Static value-sensitive factors can actually be measured and assessed at the very beginning, enabling NRT computational results.
   b. We can incorporate the dynamic value-sensitive factors into our formula at a later time because they do not play a role in the criticality of an asset unless they have specific values, and their values change frequently. We also want to make the calculations as NRT as possible. In fact, for more time-effective results, we may want to only check the assets that are being targeted for specific values.

This approach enables:

- Individual commands that know their assets best to score and rank them according to their own criteria providing accurate ranking within a command.
- Comparison of a highly critical asset in one command to a highly critical asset in another command. Since assets are rated by the decision maker in each command, two assets in different commands that have the same AC value cannot necessarily be considered to have the same criticality level despite having the same score. The weighting of the commands enables these two assets to be compared in proportion to their contribution to the upper level mission.
- Dependencies among assets within the same command or in different commands to be captured and factored into the AC metric.
- The different steps to be pre-calculated and done in parallel to provide NRT results. When non-subjective factors are used, the computation of asset criticality scores can be done with limited human interaction as well.

We have mentioned that the decision-maker at the level most knowledgeable about a set of assets would score these assets, and that command-level decision-makers would rate the appropriate commands. This will be achieved through a survey-type mechanism that would include questions about various aspects (i.e. factors) of the assets and/or commands. The responses to the survey would be used to identify relevant factors and their weights for each asset and command, as well as information about the dependency relationships. Then the proper value range and unit of measurement for each factor would be determined, and the results normalized before entered into a decision matrix. These steps are outside the scope of this paper. The techniques in this paper assume that input from decision-makers has somehow been received, if necessary, converted into a numerical format, and are ready to be used in the AC metric computation.

4. Potential Methods to Compute Asset Criticality

Some work has been done to determine cyber asset criticality [**8**] [**9**] [**10**]. These approaches either view asset criticality from the prospective of the attacker or only focus on the dependency aspect of asset criticality. Camus [**8**] is a system that provides the context to support automated mission impact assessment. It uses an ontology-based approach to integrate and fuse data from disparate locations and formats. While it provides in interesting approach, it still requires human decision makers to make a determination regarding the criticality of an asset, using the information presented to them. Sawilla et al. focus on the criticality of an asset from an attacker's point of view using dependency attack graphs [**10**] rather than focus on the impact of the asset itself. Their approach uses a generalization of Google's PageRank algorithm [**11**] to calculate the importance of an asset to an attacker. Beaudoin et al. also suggest the need for appropriate asset ranking [**9**]. Their approach is to model user services with systemic dependencies to assess the value of network assets as well as determine which user services depend on them. These approaches do not provide a comprehensive view of an asset's criticality and do not take into consideration the unique organizational structure of the DoD. As mentioned earlier, we determine asset criticality in terms of its importance to the mission, particularly in the DoD environment. We believe that the other approaches provide a complimentary method to ours.

There are numerous methods in the literature for ranking/scoring/rating items. These methods stem from various branches of science such as decision making, preference ranking, artificial intelligence, and web search algorithms. While these methods may be suitable in particular domains, the applicability of these methods to our purposes and environment need to be examined. Applicability can only be determined once we understand the environment and characteristics in which these methods are being used. In this section we briefly review some of the interesting approaches.

4.1 Various Ranking/Scoring Methods

1) Decision Making

Decision-making can be defined as the process of selecting the best alternative (or course of action), out of a set of possible alternatives to achieve a specific goal. In our daily life we make decisions on a regular basis – selecting what clothes to wear, what to eat for lunch, buying a car, etc. Each decision is made (subconsciously or not) by analyzing a finite set of alternatives against a set of criteria. In the decision making process, the final goal is to rank all the alternatives against the goal, pick the best alternative, or to determine the relative total priority of each alternative. Multiple-Attribute Decision Making (MADM) is a well-known branch of operations research models that deals with decision problems under the presence of a number of decision criteria [**12**]. The MADM approach requires that the selection be made among decision alternatives described by their attributes/criteria. For example, buying a car may require looking at different models (i.e. alternatives) with respect to the various attributes/criteria they would be compared against (e.g., price, gas mileage, color). In MADM, the decision space is discrete, meaning that a finite number of alternatives and attributes need to be assessed. In particular, MADM examines a set of problems where the goal is to find the best solution among all feasible alternatives according to the assessment of multiple quantitative and qualitative attributes.

These techniques need not be limited to decision making. When the goal is not just to rank the alternatives or pick the best one but to determine relative priority, these concepts can be applied to the problem of determining criticality of the assets for cyber network defense. Instead of deciding which alternative is the best out of all the candidates, the process would be used to facilitate determination of the most critical assets. For our purpose, alternatives are the assets that are being protected/monitored by cyber warriors and attributes are the factors/criteria we use to determine how important the asset is to the criticality of the overall mission. The result is an asset criticality (AC) metric and ranking for each asset.

Most decision making methods are interested in selecting the optimal choice, so their focus is more on the ranking of each alternative than the scores that the alternatives receive from the decision making process. However, in determining the criticality level of an asset, we need not only the ranking but also the scores that represent the *level of criticality* for each asset. The calculated scores should reflect the relative importance to the goal, or in our case, the relative importance of the asset to the overall mission. In addition, the dynamic nature of the military requires that attribute values of the assets may change frequently or that assets may be added to or removed from a network. When these situations occur, asset scores should not have to be recalculated across the board. Otherwise, it will be too time-consuming and labor intensive. Therefore, if this approach is used, we need to find algorithms that fit our criteria or adapt them in a manner that enables us to use them.

2) Recommender Systems

Companies such as Netflix and Amazon use recommendation algorithms to predict and recommend items of interest (books, digital media, etc.) to the user [**13**]. In general, recommender systems compare a user profile to some reference characteristics and attempt to predict the 'rating' that a user would give to an item. These characteristics may come from the information item (content-based approach) or the user's social environment (collaborative filtering approach) [**14**]. Used in combination with a case-based reasoning approach, it seems that a recommender system could be used to 'predict' the rating of an asset by looking at similar assets (case-based reasoning) and ratings of previously scored assets (recommendation systems). However, in our case, we do not have any historical data to use as a test case to base the predictions on. Also, the dynamic nature of DoD computer systems prohibits such a static approach. Furthermore, it is an ongoing process in which the predictions become more accurate as more items are rated. In our environment, we cannot constantly have user's rate assets and continually 'tweak' our ratings based on continued input from the user, particularly since one of our objectives is to offload decision making from the human. Furthermore, recommendation systems are based on user profiles. Either predictions based on ratings from similar users (e.g. k-nearest neighbor) or *users that like X also like Y-type* predictions are made to suggest ratings of items. This does not fit well with an asset criticality model since asset ratings should not be based on the preference or rating history of the decision-maker.

3) Various AI-related techniques

Neural networks, learning algorithms and various factor analysis techniques (Q factor, R-factor, Principal component analysis, etc.) fall into this category. While they provide interesting approaches, and may be useful in a small section of the approach, in general, they do not fit our model of hierarchical islands of assets and islands of dependencies.

Factor analyses are statistical methods used to describe variability among observed variables in terms of a potentially lower number of unobserved variables called factors [**15**]. In particular, factor analysis estimates how much of the variability of a variable is due to common factors while principal component analysis (PCA) takes into account all the variability in the variables. While there is some limited value in these approaches for determining which variables (i.e. factors) may affect asset criticality the most, they require some historical data to compare to. Neural networks are generally used for non-linear statistical data modeling to model complex relationships between inputs and outputs or to find patterns in data [**16**]. One requirement of autonomous systems is that they be able to provide justification as to the decisions made, so that human decision-makers can determine how the system reached a certain conclusion. Neural networks are generally too complex to provide supporting evidence that can be presented in a meaningful way to a human decision-maker.

4) Risk Management Approaches

By definition, a risk is a potential problem that the system or user may experience. Risk analysis techniques are frequently used to compare the value of an asset against the likelihood of a realized threat and potential losses, to determine whether or not to protect an asset [**17**] [**18**]. This type of approach is more suitable for the commercial sector where the risk does not lead to loss of lives and so a cost-benefit analysis can be performed. We are more interested in creating a standard framework for determining the value of the assets. Risk management techniques generally view risk in terms of consequences. We want to determine the criticality of assets independent of the potential risks or threats to it, and because of its value (direct or not) in the organization. Furthermore, risk analysis techniques are generally too simplified and one-dimensional for determining asset criticality in cyber network defense. Many risk analysis procedures tend to identify the vulnerabilities and risks first, and determine potential losses from these issues based on probabilities. Identification of critical assets is performed after this cost-benefit analysis. For proper course of action determination, identification of critical assets is the basic, first thing to be done. And by identification, we mean not only naming them but assessing their value in the overall scheme – in other words, how critical are they. Only after we know which assets are critical and how critical they are, can we understand the consequences of losing these assets, and produce a risk management strategy in which actions to deter threats or mitigate vulnerabilities can be correctly applied.

There has also been (slightly) similar work done from the US national critical infrastructure protection act of 1996 [**19**]. They ‘identify’ various critical sectors and assets within these sectors. However, the methodology used to determine the criticality of an asset is not standardized or methodical. The DoD requires decision networks and decision support tools that provide an explanation of the decision making process so that human decision-makers can understand the logical process that enabled the autonomous decision tool to reach its conclusion.

5) Hierarchical Decision Making Algorithms

Judgments/decision making with respect to asset criticality for the DoD must be hierarchical in nature. That is because there is no one person who has knowledge of all the assets. Eventually, there may be software that can be used across the DoD that collects parameters for each machine. The data from this software may be used to determine the criticality of an asset. Even then, there may be qualitative criteria whose value can only be determined by various decision-makers in the respective commands. Therefore, hierarchical decision-making needs to be factored into the asset criticality algorithm. The literature presents some hierarchical decision-making in the decision sciences area that may be applied to our situation. In particular, Analytic Hierarchy Process (AHP) [**20**] and Hierarchical TOPSIS [**21**] [**22**] are some examples. However, these approaches generally assume one methodology is used throughout the hierarchy and that the alternatives (i.e. in our case the assets) are all located at the bottom of the hierarchy. This is not suitable for our environment, since in a DoD command structure, assets are located throughout the hierarchy, and not just at the bottom level. If a hierarchical decision making approach is used for the hierarchical aspect of our algorithm, we will need to adapt it to fit our environment.

We have briefly discussed the pros and cons of different possible approaches to calculating asset criticality. The followings are additional potentially useful methods, but focusing on the dependency aspect of asset criticality.

6) Google's PageRank algorithm

Search engines deliver relevant search results to the user using a variety of (proprietary) algorithms that measure the relative importance of a page. Google's PageRank algorithm sorts the results of a query by the most relevant/important that match a given search string so that indexed pages can be listed in order of importance, making it easier for the user to find pages relevant to their search parameters [**23**]. PageRank assesses the importance of a web page by the number of pages linking to it as well as the importance of these pages. The straightforward and intuitiveness of the algorithm has made it popular to apply in other areas such as attack paths [**10**] and measuring species' importance of co-extinction [**24**]. Very simply stated, the algorithm recursively calculates a page rank value in which for each page that links to it, the page rank value of that page is divided by the number of outgoing links from that page, and then these values are summed up [**11**]. At first glance, the algorithm to assess the importance of web pages seems quite applicable to assessing the criticality of assets. In particular, the PageRank method of associating importance with the number of important pages pointing to it seems an interesting way to rank the dependencies of assets as a way of determining asset criticality. After all, it seems intuitive that if many (critical) assets depend on one asset, it follows that the provider asset should also be critical. However, PageRank and other similar algorithms all model resources that have different characteristics than ours. Theirs are a group of web pages and ours are a group of computers. Web pages can be reached even without links to them as long as the URL is entered into the browser. Computer assets that have no links to them are not connected to other assets[2], resulting in isolated islands of dependencies which cannot be modeled with a PageRank-type algorithm. For example, while the dependency aspect of criticality should reflect the number of assets depending on it, as well as the criticality value of the dependent assets, dependency cannot be modeled as a 'sum' of these values as page rank is. In fact, the strength of the dependencies also needs to be factored into the model. Furthermore, unlike the PageRank algorithm, just because node A has two outbound links to two separate nodes B and C (i.e. dependent on B and C for input), the criticality value of A is not 'split' between B and C. In short, the underlying assumptions made regarding the importance of web pages and other

[2] In our case, we are not talking about physical links or connections, but rather directional links between two nodes that represent a dependency relationship between the nodes where one relies on the other for input to complete its mission.

systems with similar characteristics cannot be applied to computer systems when trying to measure asset criticality.

7) Graph Theory and Network Analysis

Network analysis based on graph theory provides other means of measuring the centrality of a vertex within a graph to determine the importance of the node within a given graph. These approaches have been applied to social networking, urban network planning, space syntax, etc. to identify key entities/individuals in a network [**25**]. The widely used measures of centrality are degree centrality, betweenness, closeness, and eigenvector centrality [**26**]. In fact, the PageRank algorithm mentioned previously is a variation of eigenvector centrality. Degree centrality is defined as the number of links associated with a node. In networks, this can be applied to the degree/level of risk for the node catching a virus that is flowing through a network. While it pertains to one aspect of our dependency relationship, it does not consider the strength of the dependenices or the values of the dependent nodes. Betweenness is a measure of a vertex within a graph, i.e. the extent to which a node lies between other nodes in the network. This measure takes into account the connectivity of the node's neighbors, giving a higher value for nodes which bridge clusters. The measure reflects the number of people that a person is connecting indirectly through their direct links. Thus nodes that occur on many shortest paths between other vertices have higher betweenness than those who do not. It does not fit with our model. Closeness is the degree an individual is near all other individuals in a network (both direct and indirect links). It reflects the ability to access information through the network 'grapevine' and is equivalent to the concept of shortest path. Closeness also does not fit our model. Lastly, the eigenvector centrality measures the importance of a node in a network by assigning scores to all nodes based on the principle that connections to nodes having a high score contribute more to the node in question, as seen in the PageRank algorithm previously discussed. As mentioned in the PageRank discussion earlier, eigenvector centrality does not fit our model either, due to the disconnected islands of dependencies and the fact that the criticality of an asset cannot simply rely on the 'sum' of other nodes connecting to it. While these various measurements are useful in many applications, they do not fit well with our definition of asset criticality. The level of criticality of an asset cannot be accurately measured using degree, betweenness, closeness, or eigenvectors. Whatever approach used needs to be compatible with our other approaches and be combined in a way that enables asset criticality values to accurately reflect the true mission importance of an asset.

8) Bayesian Networks

A Bayesian network is a network-based framework for representing and analyzing models involving uncertainty. They can be used to represent probabilistic relationships between (conditionally) independent random variables. They are useful for modeling decision aids, data fusion, medical diagnostics, and others [**27**]. Graphically, they are directed acyclic graphs whose nodes represent random variables and edges represent conditional dependencies. Each node is associated with a probability function that takes as input a particular set of values for the node's parent variables and gives the probability of the variable represented by the node. Potentially, we may be able to model the dependency aspect of the AC metric by using Bayesian networks to answer the question "What is the probability that the given asset is a critical asset, given that its dependent nodes are also critical assets?" However, Bayesian networks are directed, acyclic graphs, and our model is not limited to acyclic dependency relationships. Also, in modeling dependency relationships, we are not dealing with uncertainty. In other words, we are not determining the probability of a provider node being critical given that its dependent nodes are critical. We know that if many critical assets depend on a provider asset, then the provider asset is also critical. We just don't know the level of criticality to assign to it. Therefore, Bayesian networks are not suitable for our approach.

So far, we have provided an overview of various approaches that could be used to measure asset criticality. Of these approaches, the most promising and useful ones are the MADM-based approaches. They can be applied to steps 1 and 2 of our algorithm in section 3. In the following subsection we provide some detail on a couple of well-known and well-used MADM algorithms as background.

4.2 MADM Methods

The MAMD problem can be concisely expressed in a matrix format where rows represent attributes and columns list the competing alternatives (candidates). Thus, a typical element x_{ij} of the matrix indicates the performance rating of the j^{th} alternative, A_j, with respect to the i^{th} attribute C_i. This matrix format is called a decision matrix [**28**]. The attributes, which are our factors for the purpose of asset criticality determination, are not necessarily equally important making the determination of appropriate weights a prime concern. Furthermore, attributes can have different units of measurement that need to be homogenized through a normalization procedure. In this section, we examine two different MADM decision-making methods, Simple Additive Weighting (SAW) [**29**] [**12**] and Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [**12**] [**30**]. They were selected here because of their seeming applicability to the asset criticality problem, as well as being well-used, well-known methods in the field.

SAW (Simple additive weighting) method

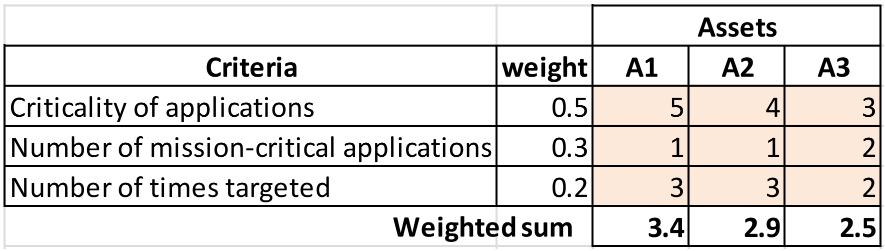
This is the most commonly used and straightforward MADM method and is also known as the weighted sum method. Each alternative is assigned a weight of which the sum of all the weights is equal to 1. Each alternative is assessed regarding its overall performance across all attributes in the following way:

$$\varphi(A_j) = \sum_{i=1}^{m} w_i z_{ij}, \qquad i = 1, \dots, m$$

Where z_{ij} is the normalized value of each element x_{ij} in the decision matrix. Of the possible values from the above formula, the decision-maker chooses the alternative A^* such that $\varphi(A^*) \geq \varphi(A_i)$ for all i. For asset criticality, factors that determine asset criticality would be multiplied by their weights and summed. This would result in asset criticality scores in which A^* is the most critical asset. Table 1 shows a decision matrix composed of three assets, A1, A2, and A3, with values ranging from 1 to 5. We can see that A1 is the most critical asset with a score of 3.4 when using the SAW method.

The advantage of this method is that it is a proportional linear transformation of the raw data which means that the relative order of magnitude of the standardized scores remains equal. This means that the ratio of scores among assets is proportional to the relative criticality.

Table 1. Sample calculation of asset criticality using SAW



		Assets		
Criteria	**weight**	**A1**	**A2**	**A3**
Criticality of applications	0.5	5	4	3
Number of mission-critical applications	0.3	1	1	2
Number of times targeted	0.2	3	3	2
Weighted sum		**3.4**	**2.9**	**2.5**

TOPSIS Method (Technique for Order Preference by Similarity to Ideal Solution)

This approach ranks alternatives according to their closeness to a hypothetical positive ideal solution (zenith) and a hypothetical negative ideal solution (nadir). The domain set of alternatives is defined as a m-dimension Euclidean space. Therefore, each alternative is represented as a point in this space. A basic assumption made here is that each attribute is characterized by monotonic increasing or decreasing utility. Using the TOPSIS principle, the solutions (i.e. most preferred alternatives) are those which are at the same time farthest from the nadir point and closest to the zenith point, with distance being measured by Euclidean distance. TOPSIS models this principal by defining 'relative closeness to the ideal solution', T_j^* according to the following relation:

$$T_j^* = \frac{D_j^-}{D_j^- + D_j^+} \tag{1}$$

Where D_j^- is the m-dimensional Euclidean distance between the j^{th} alternative and the nadir point, and D_j^+ is the m-dimensional Euclidean distance between the j^{th} alternative and the zenith point. According to this method, the alternative A_j is better than A_m if $T_j^* > T_m^*$ or $D_j^- / \left(D_j^+ + D_j^-\right) > D_m^- / (D_m^+ + D_m^-)$. This calculation is necessary because there can be many alternatives that are closest to the ideal solution. Also the alternative closest to the ideal solution is not necessarily the farthest from the negative-ideal solution. As can be seen in Figure 3, if we only have alternatives A_1, A_2, and A_3, they are all equidistance from A^*, the positive-ideal solution, but have different Euclidean distances from A^-. Now, if we consider A_1 and A_4, A_4 is closer to the ideal solution, but A_1 is farther from the negative-ideal, A^-. The relative distance measurement handles these conflicts.

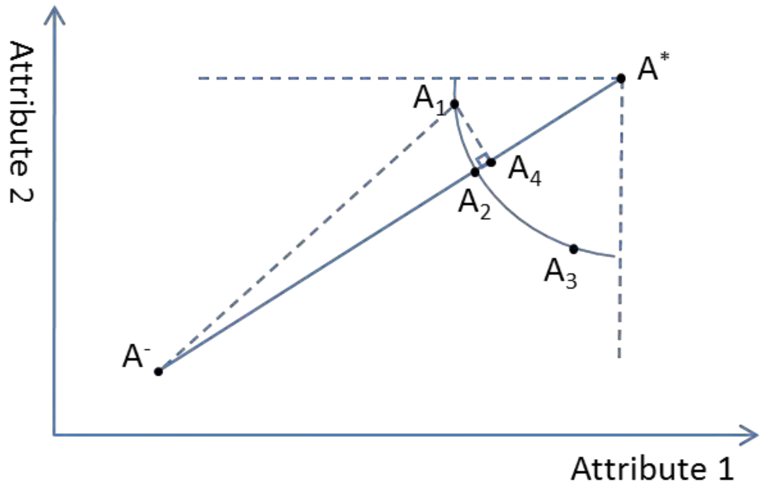


Figure 3. Euclidean distances to positive and negative ideal solutions

In order to calculate the relative closeness to the ideal solution, the normalized decision matrix is first calculated using the vector method:

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{ij}^2}}$$

where x_{ij} is the value of alternative j with respect to attribute i $(1 \leq i \leq m, 1 \leq j \leq n)$. Then, the weighted normalized decision matrix is calculated as:

$$v_{ij} = w_i z_{ij}, \qquad i = 1, \ldots, m, j = 1, \ldots, n$$

Where w_i is the weight of the i^{th} attribute. The zenith A^* and nadir $A^-$are calculated as follows:

$$A^* = \{v_1^*, \ldots, v_m^*\} = \{(max_j v_{ij} | i \in I'), (min_j v_{ij} | i \in I'')\},$$

$$A^- = \{v_1^-, \ldots, v_m^-\} = \{(min_j v_{ij} | i \in I'), (max_j v_{ij} | i \in I'')\},$$

Where I' is associated with benefit criteria, and I'' is associated with cost criteria[3]. Thus, the zenith (positive ideal solution) is made up of the best value of each criterion over all attributes and the nadir (negative ideal solution) is composed of the worst value of each criterion over all the attributes. The respective Euclidean distances are calculated as:

$$D_j^+ = \sqrt{\sum_{i=1}^{m} \left(v_{ij} - v_i^*\right)^2}, \qquad j = 1, \ldots n$$

$$D_j^- = \sqrt{\sum_{i=1}^{m} \left(v_{ij} - v_i^-\right)^2}, \qquad j = 1, \ldots n$$

[3] A benefit attribute is one in which higher measures are more desirable for the decision-making problem, and a cost attribute is one in which lower measures are more desirable. For example, when buying a car, the gas mileage of the car is a benefit attribute, and the price of the car is a cost attribute.

The relative closeness is then calculated as in equation (1) to obtain the score for each alternative, after which the alternatives are ranked. An example of applying TOPSIS to the same assets as those used in the previous example is provided in Table 2. The cells highlighted in yellow are the best value of each criterion over all the attributes, and together compose the zenith. The cells highlighted in purple are the worst value of each criterion and compose the nadir. In other words, the zenith is $A^* = \{2.5, 0.6, 0.6\}$ and the nadir is $A^- = \{1.5,0.3,0.4\}$. The distances to the zenith and nadir are computed using these values against each value in the alternatives. For instance, the distance to A^* from A1 is $D_1^+ = \sqrt{(2.5-2.5)^2 + (0.6-0.3)^2 + (0.6-0.6)^2} = 0.30$ and the distance to A^- from A1 is $D_1^- = \sqrt{(2.5-1.5)^2 + (0.3-0.3)^2 + (0.6-0.4)^2} = 1.020$. The resulting relative closeness value T_1^* for A1 is $1.020/(0.300 + 1.020) = 0.773$. Alternatives A2 and A3 are calculated in a similar manner. Again, the most critical asset is A1 with a score of 0.773, which is its relative closeness to the zenith.

Table 2. Sample calculation of asset criticality using TOPSIS

		Assets					
Criteria	**weight**	**A1**	**A2**	**A3**	**value x weight**		
Criticality of applications	0.5	5	4	3	2.5	2	1.5
Number of mission-critical applications	0.3	1	1	2	0.3	0.3	0.6
Number of times targeted	0.2	3	3	2	0.6	0.6	0.4
		Distance to Nadir			0.300	0.583	1.020
		Distance to Zenith			1.020	0.539	0.300
		Relative Closeness			**0.773**	**0.480**	**0.227**

Eigenvector Prioritization Method for Obtaining Weights

In order to apply a decision-making method, we need to calculate the relative importance of each criterion, i.e. their weights. Weights enable decision-makers to specify the importance of each attribute, relative to others, since not all attributes are likely to be considered equally important. This process is generally subjective, requiring input from the decision-maker and is likely to vary from decision-maker to decision-maker. The decision-maker can express his preferences in either an ordinal or cardinal scale, although most models require cardinal weights that are then normalized to sum to 1, that is, $\sum w_i = 1$ where w_i is the weight assigned to the i^{th} attribute. Various weight assignment techniques are available such as assigning weights from ranks and ratio weighting. Rank-based weighting requires ranking all the attributes at the same time, placing a heavy cognitive burden on the decision-maker [**12**]. This makes ratio-based weight schemes such as Saaty's Eigenvector Prioritization method [**31**] [**30**] more preferable. In this method the decision-maker performs a pairwise comparison of the criteria that they choose for their assets. Pairwise comparison involves comparing two attributes at a time, asking the importance ratio between them. Mathematically, for n attributes, we would need (n-1) pairwise comparisons. However, the result of comparing how much more important C1 is than C2 is the inverse of comparing how much more important C2 is than C1. Also, the value of an attribute compared with itself is always 1. Therefore, only $n(n-1)/2$ pairwise comparisons are needed. In this method, a pairwise comparison matrix is constructed using a scale of relative importance with the decision maker's preferences being stored in either the lower or upper half of the matrix. Saaty uses a scale of 1-9 with 1 being equivalent and 9 being absolutely important. The type of scale used can be adjusted to fit the application. Then, the geometric mean of each row is computed, then normalized to obtain the final weights of the criteria. For example, given three criteria C1, C2, and C3 the decision-maker will construct a comparison matrix that may look like that of Table 3a [**12**]. The geometric mean and resulting normalized weights would be computed as in Table 3b.

Table 3a. Pairwise comparison matrix

	C1	C2	C3
C1	1.0	0.3	0.5
C2	3.0	1.0	3.0
C3	2.0	0.3	1.0

Table 3b. Weight computation

	Geometric mean		*Weight*
C1	(1x1/3x1/2)$^{1/2}$ = 0.5503		0.1571
C2	(3x1x3)$^{1/2}$ =2.0801	=	0.5936
C3	(2x1/3x1)$^{1/2}$ = 0.2493		0.2493

5. Our Approach

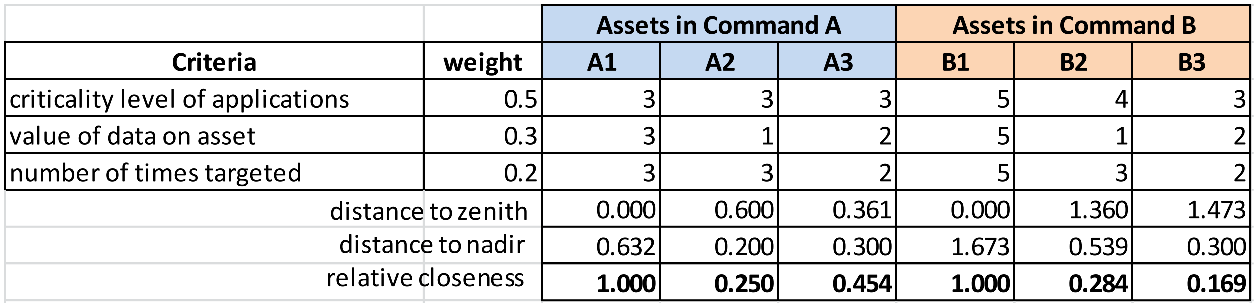
We have introduced the high level overview of our approach in scoring and ranking assets in Section 3. Each step of the approach is expanded in more detail below.

Step 1. Calculate AC scores based on static factors using a decision-making algorithm
We examined SAW and TOPSIS because of their popularity and applicability. Neither of these algorithms is suitable as-is to use because of our special requirements in section 3, but both possess pros and cons. Particularly, we do not want to recalculate scores for all assets when the value of an asset changes, or a new one is introduced or deleted. Also, we want to compare across scores regardless of which method was used. While each method provides relative scores, the different calculation methods lead to different characteristics; the SAW method uses the overall value of each alternative to compute a score, while TOPSIS uses distance information. Also, the required calculation time for each is different. Since SAW calculations are done independently of other assets, if the value of an attribute in an asset changes only that one asset needs to be recalculated. In addition, when an asset is added or removed, it does not affect the scores of the other assets.

On the other hand, TOPSIS calculates its scores across all other alternatives using the concept of positive and negative ideals, In particular, the positive ideal and negative ideal in TOPSIS are relative values, calculated using the given values in the decision matrix. The use of relative values for positive and negative ideals is particularly problematic for our dynamic environment where asset scores can change, assets can be added or removed, and assets need to be compared across different commands. When a change in an asset's value affects the positive or negative ideal, all assets' distances to the positive and/or negative ideal (and thus the relative closeness) need to be recalculated. For example, going back to the example given in Table 2, assume that the value for 'Number of mission-critical applications' changed from 1 to 3 for asset A1. The resulting positive ideals and negative ideals would become $A^* = \{2.5, 0.9, 0.6\}$ and $A^- = \{1.5, 0.3, 0.4\}$, respectively. The negative ideal has not changed, but the positive ideal has, requiring recalculation of all assets' distances to the positive ideal as well as their relative closeness values.

Furthermore, this approach makes the positive and negative ideals different for each command, leading to incompatible relative distance values. This problem is illustrated in Table 4.

Table 4. Problems with positive and negative ideal values in TOPSIS



		Assets in Command A			**Assets in Command B**		
Criteria	**weight**	**A1**	**A2**	**A3**	**B1**	**B2**	**B3**
criticality level of applications	0.5	3	3	3	5	4	3
value of data on asset	0.3	3	1	2	5	1	2
number of times targeted	0.2	3	3	2	5	3	2
distance to zenith		0.000	0.600	0.361	0.000	1.360	1.473
distance to nadir		0.632	0.200	0.300	1.673	0.539	0.300
relative closeness		**1.000**	**0.250**	**0.454**	**1.000**	**0.284**	**0.169**

The table shows two hypothetical commands. Command A does not have any important assets in the sense that its most critical asset only has medium-grade values (i.e. all 3s). However, because the positive ideal and negative ideal solutions are composed from the best values across all alternatives, A1 has a relative closeness value of 1. Likewise, a highly critical asset, B1, in command B also has a relative closeness value of 1. It is safe to assume that A1 and B1are not equally critical to the overall mission, and the rating of the commands will help differentiate these scores to a degree. But essentially, the problem lies in the fact that each command's assets are rated with respect to the relative values of the assets within the command. While A1 should still be ranked and valued as more critical than assets A2 and A3, it should not be held up as the 'ideal' asset, as the TOPSIS method currently has it.

In general, both SAW and TOPSIS provide similar rankings but TOPSIS has better distinguishing capabilities, delineating the differences between similar alternatives [**32**] [**33**] [**34**]. For example, the decision matrix in Table 5 shows three alternatives compared using three criteria/attributes that are all identical in importance. Scores are computed for the alternatives using both the SAW and TOPSIS methods. The three alternatives would be equally preferred using the SAW method. Borrowing from utility theory, we would say that the decision-maker is indifferent between the three alternatives – i.e. all three are equally critical. However, the results of TOPSIS show that the relative distance measurements of the three assets are not identical. TOPSIS provides some differentiation between the three alternatives. Using this example, we argue that assets are not equally critical just because their values add up to identical sums and that TOPSIS results reflect this.

Furthermore, because of its simple calculation method, SAW is mostly used as a method to obtain a preliminary outcome [**29**]. For these reasons, we selected TOPSIS as our initial method, but modified it slightly to overcome the limitations discussed above.

Table 5. Comparison of SAW and TOPSIS

		Assets					
Criteria	**weight**	**A1**	**A2**	**A3**	**value x weight**		
Number of mission-critical applications	0.33	4	3	5	1.33333	1	1.66667
Criticality of applications	0.33	4	5	5	1.33333	1.66667	1.66667
Number of times targeted	0.33	4	4	2	1.33333	1.33333	0.66667
				SAW	**4**	**4**	**4**
				Relative Closeness	**0.613**	**0.528**	**0.528**

The problem with using TOPSIS in our environment stemmed from the way positive and negative ideal values were selected. We modify the distance measurement calculation so that *absolute* positive and negative ideal values are used as the new zenith and nadir, respectively. In other words, we create a hypothetical asset that has all possible maximum values and a hypothetical asset that has all possible minimum values, and calculate the distance to the zenith and nadir from these values respectively. For example, using the decision matrix given in Table 2, the new absolute zenith is $A^* = \{2.5, 1.5, 1.0\}$ and the new absolute nadir is $A^- = \{0.5,0.3,0.2\}$ after weights have been applied. Using these values to calculate distances and then relative closeness measurements, the new TOPSIS scores would be that of Table 6.

Table 6. Calculation of asset criticality using modified TOPSIS

		Assets						Abs. Pos.	Abs. Neg.
Criteria	**weight**	**A1**	**A2**	**A3**	**value x weight**			**ideal**	**ideal**
Criticality of applications	0.50	5	4	3	2.5	2	1.5	2.5	0.5
Number of mission-critical applications	0.30	1	1	2	0.3	0.3	0.6	1.5	0.3
Number of times targeted	0.20	3	3	2	0.6	0.6	0.4	1	0.2
		Relative Closeness			**0.617**	**0.533**	**0.419**		

Unlike the original TOPSIS approach that uses only the values available in the decision matrix, these zenith and nadir values are fixed. The advantage of this is that if an asset's value changes or the asset itself is removed from the decision-making process, the zenith and nadir values do not change, and all the sub-calculations for relative closeness do not need to be recalculated for each asset: if an asset's value changes, then only that asset's relative closeness (to the absolute positive and negative) needs to be recalculated. If an asset is removed, no recalculation of any relative closeness values is required since the zenith and the nadir are always computed using absolute values. In addition, each command's assets are rated against their own absolute positive and negative values, so that they all scale in a consistent way. Modifying TOPSIS in this manner allows asset criticality scores to be comparable and minimizes recalculation time. In a dynamic environment, this type of approach is advantageous and necessary.

As we mentioned earlier, each evaluator/decision-maker needs to be able to assess his assets independently of other commands, using whatever criteria they deem important to determining the criticality of assets. Because each decision-maker rates assets independently of each other, the assets from different commands are not directly comparable at this stage. However, as a result, as long as assets are rated in a consistent manner and their relative importance is represented as a score, these values can be normalized for inter-command comparisons. This does not imply that the AC metric of an asset from one command can be directly compared to the AC metric of an asset from a different command. Rather, higher-level units would rank the importance of lower-level commands to provide a weight measurement that can determine how much more important one command is over another using the next step.

Step 2. Calculate the ratings for each command

The DoD command structure is hierarchical. The literature presents some hierarchical decision making models. However, mostly they assume that the same approach is used throughout the process. Some decision making models employ group decision making techniques but assume that each actor considers the same set of alternatives and criteria, which does not fit well to our problem domain. These techniques are usually applied to explore the cases of different weights (among decision makers) for each criteria, when the assessed values are objective, and/or to assess weights that are subjective, so as to reach a consensus [**20**] [**35**]. In addition, these models assume an environment such as that of Figure 4 with assets (i.e. alternatives) only at the lowest level. In reality, each command (regardless of its level in the hierarchy) possesses its own assets that need to be scored and ranked. In these cases, assets in the lower level would be unfairly penalized by their associated commands receiving a low-value weight when using the measures discussed above. Therefore, we devised a different hierarchical weight assignment scheme that does not overly penalize the lower level commands and their assets and provides a more balanced approach. This hierarchical weight assignment scheme can be implemented in two different ways. In the first weight assignment method, we ask the decision maker of each command to determine how important each of its ‘known’ commands is. By ‘known’ commands we mean the command that the decision maker belongs to, as well as the commands directly under him. This decision is made by answering the question “How much do you depend on this command to complete your mission?” For instance, the decision maker can decide that it relies on its own command 100%, while it depends on the two commands below it 100% and 80% respectively, to complete its mission. This percentage is translated down from the top level nodes to the bottom nodes to achieve a weight. For example, Figure 5 depicts a hierarchy of commands in which each decision maker has determined how much he depends on his command, as well as the ones directly under him, represented in percentages. While not shown here, we can assume that each command has assets under it that it needs to score and rank. From these values, the individual weights of the commands are calculated in a top-down manner as shown in Figure 6. The weights (given in parentheses) are calculated by taking the weight of the command directly above it, and multiplying it by dependency factor on the edge between the target command and the one above it. These weights are then used to determine the true values of the assets by multiplying the weights of each command to the AC scores of the assets within each respective command.

The second weight assignment method uses one of the MADM approaches, such as TOPSIS or SAW. As in the above method, each decision-maker rates his own command and the ones directly under him, but instead of answering a subjective question, external factors that reflect mission importance are employed. For example, criterion such as *situation at geographical location* is used to make decisions regarding importance of commands. Once each decision-maker rates its associated commands, weights are calculated the same as in the first method. A more detailed example of this method is provided in section 6.

With respect to the requirement of near real-time computation, these methods are preferable over others since they do not perform pairwise comparison across all the commands at once. A change in the dependency of one command will only change the weights of all the commands (and assets) under it, and not require total recalculation. While both methods are acceptable, the second method allows less subjective measures to be taken, while also providing the justification needed (in the form of identified criteria) to determine how a decision maker rated certain commands, as well as enabling external factors to affect mission criticality.

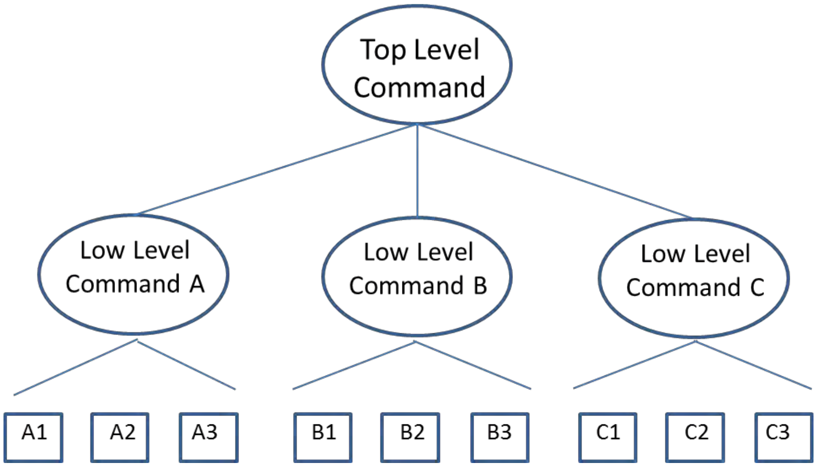


Figure 4. Commands with assets

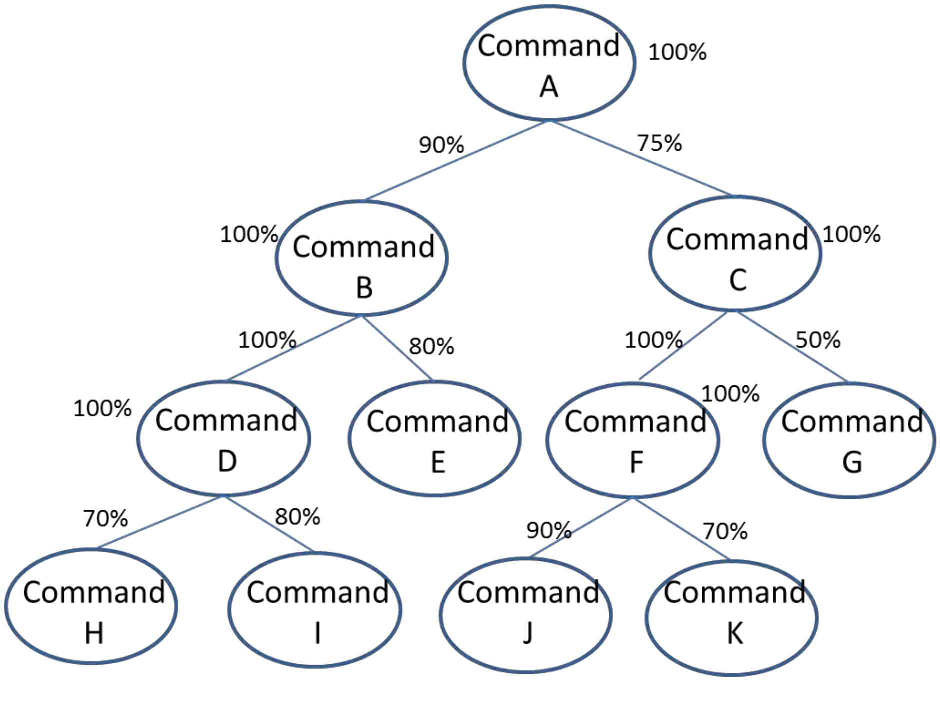


Figure 5. Hierarchy of commands with decision-makers' ratings

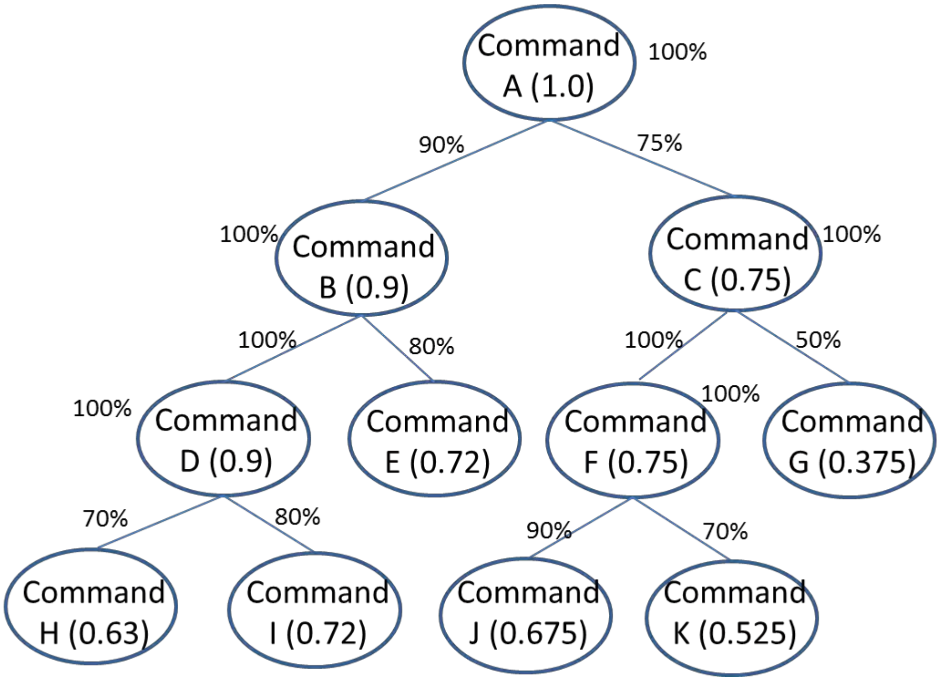


Figure 6. Weights of each command calculated

Step 3. Calculate the dependencies among assets

When computing the dependency factor of the asset, it should be a factor of how many assets depend on the given asset, the strength of the dependencies between provider asset and dependent assets, and the AC metric of the dependent assets, as well as the initial AC metric of the provider asset. Figures 7 through 9 explain this. Dependency relationships are depicted as directed graphs with nodes representing assets and arrows representing the direction of information flow in the dependency relationship from provider node (shown in blue) to dependent nodes. The numbers on the arrows represent the degree of dependency and the numbers in the nodes represent the AC metrics of the assets obtained from steps 1 and 2. Provider nodes would also have initial AC values, but to keep things simple, we do not show these values here. In Figure 7, we see two assets that have other assets depending on them. All other things being equal, the one with more assets depending on it should have a higher AC metric. On the other hand, Figure 8 shows two islands of assets, with the same number of assets depending on two different assets. The dependent assets are equally critical, but have different strengths of dependencies. The strength of dependency is a general concept that implies to what degree the dependent node relies on the provider node to complete its mission. In other words, the assets in the first group depend on the provider node 50% to complete its mission. In this case, the strength of the dependencies matters in determining which of the two provider assets are more critical. Lastly, in Figure 9, we see that all other things being equal, the initial criticality measurement of the dependent assets, denoted as integers inside the circles, determines the criticality of the provider assets. While not shown here, the initial AC metric of the provider nodes must also be taken into consideration. For example, in Figure 9, if the provider nodes had an initial AC metric of 5, then the adjusted AC metric cannot be below this, just because the dependent nodes have low AC values or low strength of dependencies. This initial AC metric was computed using other factors that determined the criticality of the asset independent of dependencies and should not be downgraded.

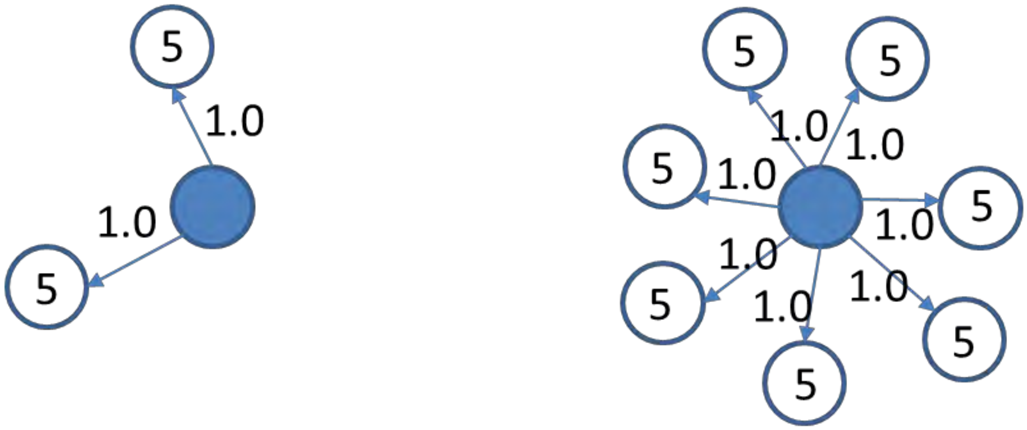


Figure 7. Comparing the number of dependencies

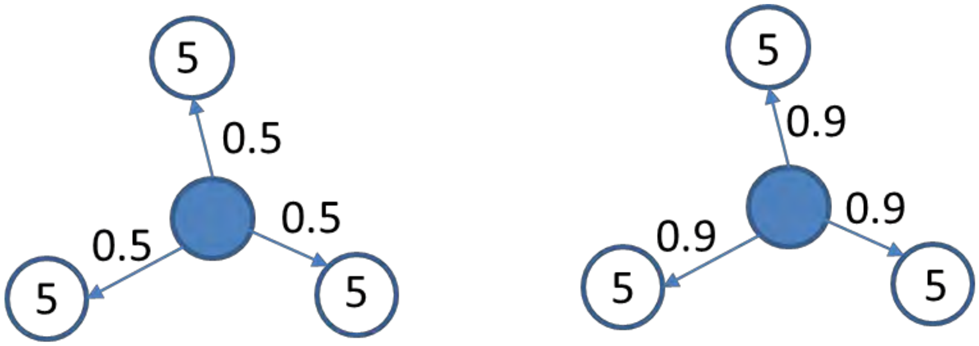


Figure 8. Comparing the strength of the dependencies

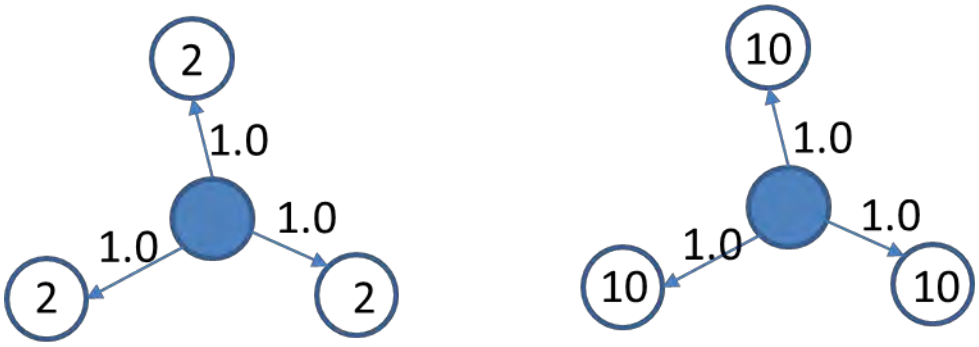


Figure 9. Comparing the AC metric of dependent assets

In summary, our dependency algorithm should possess the following properties:

- Each initial value (i.e. AC metric) of the node/assets in the dependency relationship is the independent asset criticality score obtained from the individual asset calculations computed by each command and then adjusted across all the commands using the command-level ratings.
- The AC metric of a provider asset should be a function of its initial AC metric, the AC metrics of the dependent assets, the strength of the dependencies, and the number of assets that depend on it.
- The resulting value of the providing node should have a lower bound limit of $max\{AC(a_i) \times w_i \, for\ all\ a_i \quad B_i, AC(p)\}$ where $AC(a_i)$ is the asset criticality score of the dependent asset a_i, w_i is the strength of the dependency from a_i to provider asset p, B_i is the set of all assets that depend on p, and $AC(p)$ is the original AC metric score for provider asset p. The relationship between the provider asset's value and the dependent nodes' scores, degree of dependencies or the number of dependent nodes is not linear or exponential. An asset that supports 100 other nodes likely more critical than one that supports 10 nodes, but is not necessarily 10 times more important. The criticality function would likely be defined by a logarithmic relationship between the adjusted criticality value of the provider node and the variables (i.e. AC values of dependent assets, number of dependent assets, and strength of dependencies).

While we stated the properties of the dependency algorithm above, the development of the algorithm is a work in progress. For this paper, we will just adopt the simple notion of the provider node being at least as critical as the most critical value obtained from multiplying each asset criticality value by its strength of dependency on the provider node. In other words, the new value of the provider node will be the lower bound described above.

Step 4. Rate the assets based on value-sensitive factors

Value-sensitive factors such as 'who is logged on' and 'who owns the asset' may change the criticality of an asset depending on what the value of the factor is at a given time. In most circumstances, the values of these factors do not affect the criticality of the asset, so we do not incorporate them into our initial decision-making process. However, when the factor does possess a certain value (e.g. who's logged on = Admiral), an asset with this value can then become quite critical. We assess this value at certain times depending on whether they are static value-sensitive or dynamic value sensitive factors. This part of the algorithm is not part of the paper and will be addressed separately.

6. An Example of the Asset Criticality Algorithm

In this section we will go through the various steps of the algorithm (steps 1 through 3) using a hypothetical example. In this example, we have a command hierarchy that looks like Figure 10, and each command in the hierarchy has a set of assets that need to be scored and ranked.

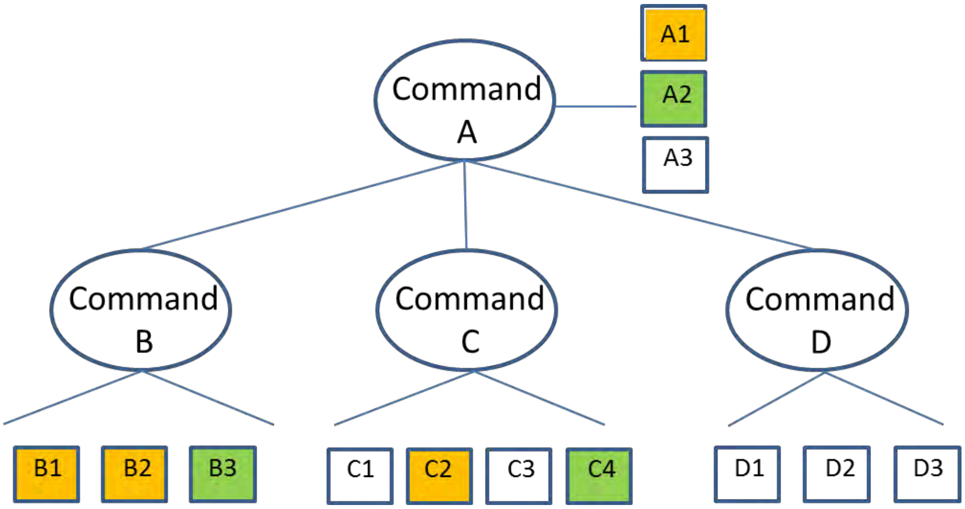


Figure 10. Hypothetical command and control structure with assets

In this scenario there are four commands, with command A as the top level command having three commands under it. Command B is a submarine, Command C is a research facility, and Command D is a medical unit. Each command has assets denoted as squares within it. Colored assets represent assets that have a dependency relationship with each other. The actual dependency relationship of these colored assets is shown in Figure 11. As before, the nodes represent the assets and the arrows represent information flow in a dependency relationship denoting the dependent node relying on the provider node for input. Labels on the arrows represent degrees of dependency with 1.0 implying 100% dependence. In other words, A1 relies on C2 100% to complete its mission, while B1 relies on C2 50% to complete its mission.

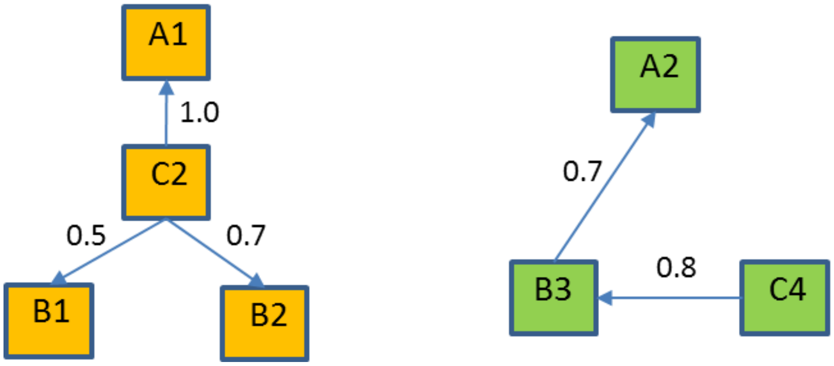


Figure 11. Network dependencies of assets

Each command will rate its assets independently of each other using some MADM approach, with the types and importance of attributes to be determined by the decision maker within each unit. The top-level command (i.e. Command A) will rate the commands under it as well as itself, and then the dependencies will be calculated.

Table 7 shows the list of potential attributes and scores for each asset. To save space, we show the decision matrix for each command in one table. While there are four factors, each command only uses three – the three factors that each command considers most appropriate to rate its own assets. As can be seen, we have assigned a numerical range of 1-5 for each factor, where 1 is low and 5 is high. For example, if there is a lot of sensitive data on the asset, then it would receive a rating of 5 for that criterion. In actuality, not all factors will have the same range of values. Thus, some normalization will be applied before calculations can be performed. But for now, we are keeping things simple.

Table 7. Decision matrix for each command

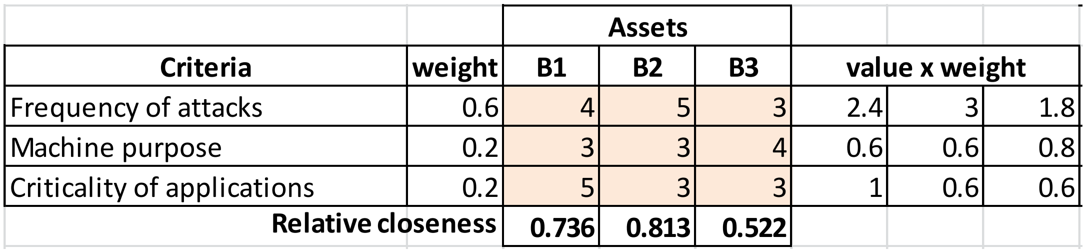
	Command A			**Command B**			**Command C**				**Command D**		
Criteria	A1	A2	A3	B1	B2	B3	C1	C2	C3	C4	D1	D2	D3
Value of data on asset	5	4	3				4	3	2	5	5	3	3
Frequency of attacks	5	4	2	4	5	3	3	3	4	4	5	2	5
Machine purpose				3	3	4	5	2	2	3	5	4	2
Criticality of applications	3	4	1	5	3	3							

We will use TOPSIS to calculate the scores for assets in each command separately. To simplify the example, weights for each factor are assigned arbitrarily. The preliminary results are as follows (Tables 8 through 11):

Table 8. Results for Command A

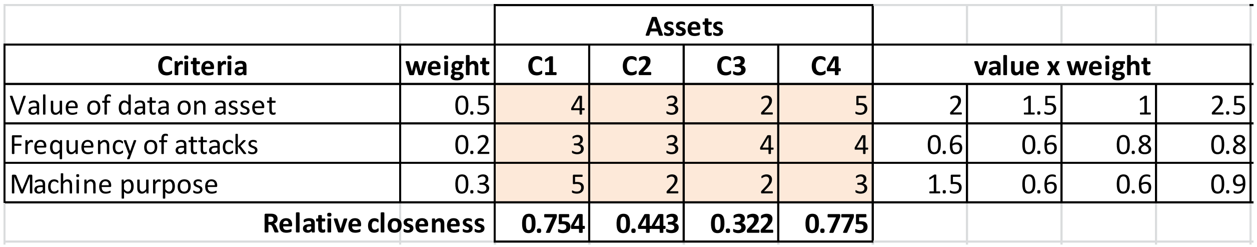
			Assets				
Criteria	**weight**	**A1**	**A2**	**A3**	**value x weight**		
Value of data on assets	0.5	5	4	3	2.5	2	1.5
Frequency of attacks	0.3	5	4	2	1.5	1.2	0.6
Criticality of applications	0.2	3	4	1	0.6	0.8	0.2
Relative closeness		**0.855**	**0.750**	**0.400**			

Table 9. Results for Command B



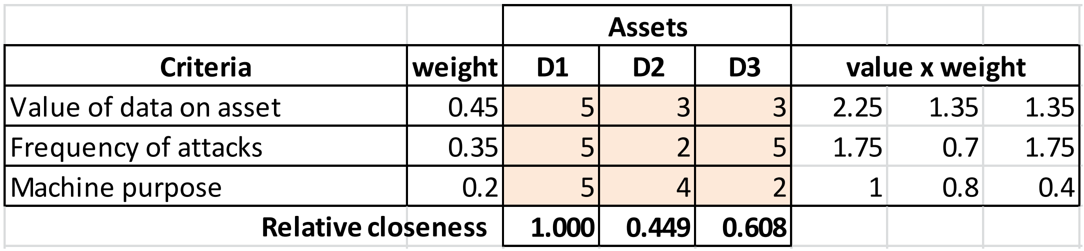
		Assets					
Criteria	**weight**	**B1**	**B2**	**B3**	**value x weight**		
Frequency of attacks	0.6	4	5	3	2.4	3	1.8
Machine purpose	0.2	3	3	4	0.6	0.6	0.8
Criticality of applications	0.2	5	3	3	1	0.6	0.6
Relative closeness		**0.736**	**0.813**	**0.522**			

Table 10. Results for Command C



		Assets							
Criteria	**weight**	**C1**	**C2**	**C3**	**C4**	**value x weight**			
Value of data on asset	0.5	4	3	2	5	2	1.5	1	2.5
Frequency of attacks	0.2	3	3	4	4	0.6	0.6	0.8	0.8
Machine purpose	0.3	5	2	2	3	1.5	0.6	0.6	0.9
Relative closeness		**0.754**	**0.443**	**0.322**	**0.775**				

Table 11. Results for Command D

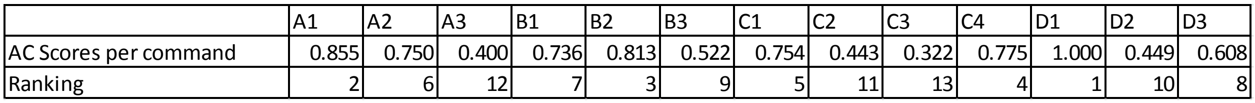


		Assets					
Criteria	**weight**	**D1**	**D2**	**D3**	**value x weight**		
Value of data on asset	0.45	5	3	3	2.25	1.35	1.35
Frequency of attacks	0.35	5	2	5	1.75	0.7	1.75
Machine purpose	0.2	5	4	2	1	0.8	0.4
Relative closeness		**1.000**	**0.449**	**0.608**			

We use our modified TOPSIS with absolute positive and negative values to obtain the results.

At this point, each decision maker from each command has rated his assets separately using different factors and weights. Even though we use a modified TOPSIS these are not comparable as is, since the ratings are based on importance to the mission of each command. Despite this, we show a ranking of the assets after completing the individual ratings in Table 12. As can be seen from the results of Table 12, asset D1, a machine in the medical unit, is ranked most critical. Whether this is truly critical or not depends on the criticality of the command itself, which is calculated next.

Table 12. Preliminary Ranking of all assets

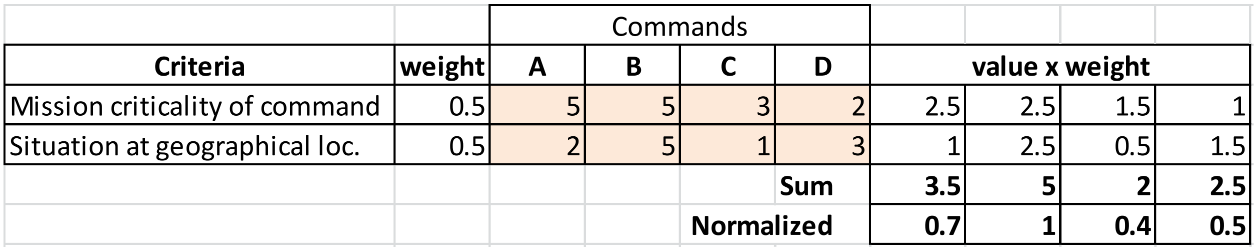


	A1	A2	A3	B1	B2	B3	C1	C2	C3	C4	D1	D2	D3
AC Scores per command	0.855	0.750	0.400	0.736	0.813	0.522	0.754	0.443	0.322	0.775	1.000	0.449	0.608
Ranking	2	6	12	7	3	9	5	11	13	4	1	10	8

Next, we need to rate the commands themselves, so that the importance of the command to the overall DoD mission can be reflected into the AC metric. As mentioned earlier, commands rate their own command and the commands directly under them. The results are flattened out as described earlier. The rating can be subjective, asking “how important is this command to the mission?” or use a MADM approach for each evaluation. While either SAW or TOPSIS would work, to minimize calculations, we will use SAW with criteria selected to determine mission importance. Command A will rate itself, as well as Commands B, C, and D using criteria and arbitrarily selected weights. The use of factors at this level enables external situations to be reflected in the decision-making process. For example, in wartime, strategic commands may have less criticality than tactical commands. Situations at geographical locations may also affect command criticality. The use of these factors enables decision-makers to appropriately reflect them in their judgment, rather than subjectively comparing commands.

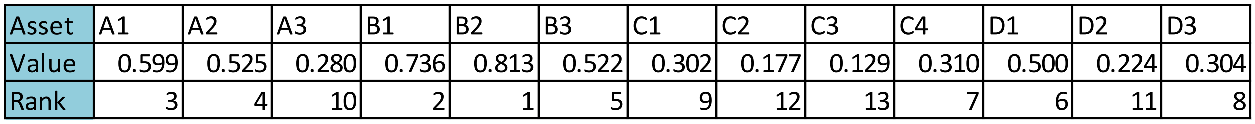
The results of the calculation (shown in Table 13) provide scores for each command that when normalized are used as weights for each asset within the respective command. For example, the normalized value of Command A is 0.7. This value is multiplied across all assets in Command A to obtain actual AC scores. Proceeding in the manner for all of the commands, we obtain the scores and ranking shown in Table 14.

Table 13. Criteria/Factors for rating commands



		Commands							
Criteria	**weight**	**A**	**B**	**C**	**D**	**value x weight**			
Mission criticality of command	0.5	5	5	3	2	2.5	2.5	1.5	1
Situation at geographical loc.	0.5	2	5	1	3	1	2.5	0.5	1.5
					Sum	**3.5**	**5**	**2**	**2.5**
					Normalized	**0.7**	**1**	**0.4**	**0.5**

Table 14. Scores and ranking of assets after command level rating applied



Asset	A1	A2	A3	B1	B2	B3	C1	C2	C3	C4	D1	D2	D3
Value	0.599	0.525	0.280	0.736	0.813	0.522	0.302	0.177	0.129	0.310	0.500	0.224	0.304
Rank	3	4	10	2	1	5	9	12	13	7	6	11	8

We can see from the results that command D, being not as important as other commands to the overall mission, has received a damping factor (i.e. weight) that lowered the scores of all its assets. Command B, which is a submarine, doing mission critical activities and situated in a wartime location is the most critical command, so it has many critical assets. Up to this point the criticality of the assets has been measured independently of other assets.

The next step is to examine dependency relationships and factor them into the criticality measure. Using the asset criticality values shown in Table 14, our dependency relationship diagram from earlier now looks like Figure 12. It is obvious that asset C2, with an AC metric of 0.177 is a much more critical asset than its results show, since three relatively critical assets depend on it for input. To calculate the new AC metric for C2, we select $max\{AC(A1) \times w_{A1}, AC(B1) \times w_{B1}, AC(B2) \times w_{B2}, AC(C2)\} = max\{0.599, 0.368, 0.569, 0.1777\} = 0.599$.

After calculations C2’s new AC metric becomes 0.599. The results for the other dependency relationship are calculated similarly. First, B3’s value is updated to be the max of A2 times its weight and B3's original value. Then C4's value is updated to be the max of B3's new value times its weight or C4's original value. The results of reflecting the dependency is shown in Figure 13 for the assets with dependency relationships, and Table 15 for all assets.

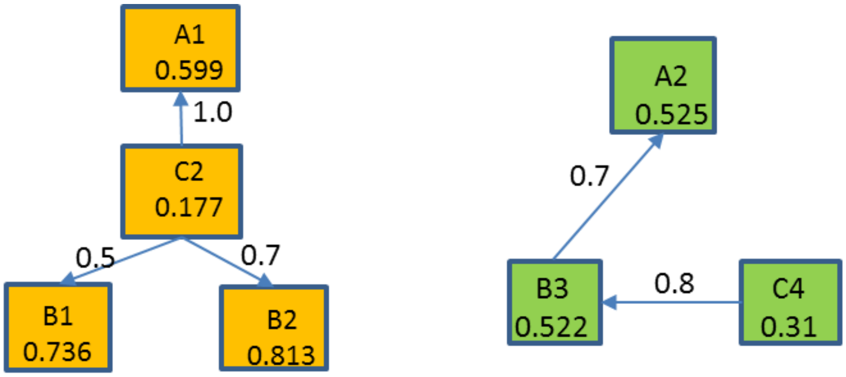


Figure 12. Dependencies with initial AC metric values

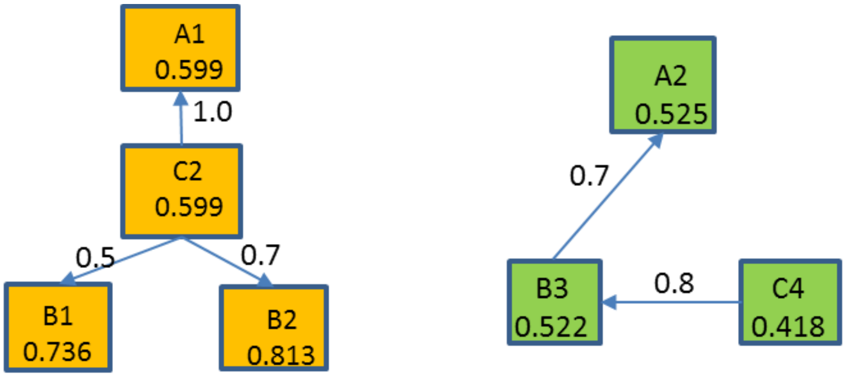


Figure 13. Dependencies with updated AC metric values

Table 15. Scores and rankings of assets after dependencies

Asset	A1	A2	A3	B1	B2	B3	C1	C2	C3	C4	D1	D2	D3
Value	0.599	0.525	0.280	0.736	0.813	0.522	0.302	0.599	0.129	0.418	0.500	0.224	0.304
Rank	3	5	11	2	1	6	10	3	13	8	7	12	9

7. Conclusion and Future Work

Proper course-of-actions should be based on the severity of an attack, the criticality of the asset it targets and the state of the asset. There are many algorithms and tools for categorizing the severity of an attack [**36**] [**37**]. However, we lack a standardized way to determine the criticality of assets, as well as the state of the asset. This paper discusses our approach to determining asset criticality in a dynamic, military environment. We outlined our four-step approach for obtaining an AC metric that can be used to determine the criticality of an asset as it relates to the overall mission. This AC metric is the missing piece in the puzzle that fits in the dynamic DoD environment where the nature of the mission may change, the definition of mission criticality can change and the values that compose a critical asset may change at any moment. Our approach is still being developed: we need to determine the important factors, how we will obtain information about these factors for each asset, what the value ranges will be, and how to normalize them. We still need to work on measuring the dependency aspect of asset criticality, as well as how to incorporate value-sensitive factors into the algorithm.

Ultimately, event information, asset criticality and asset state will need to be combined to determine proper course-of-action. In addition, the sheer amount of data makes manual aggregation of this information not feasible for vital decision-making especially in times of critical need. These limitations call for an autonomous decision-making process that takes into account the context surrounding an event for proper and timely course-of-actions.

This approach enables not only cyber warriors to make faster and more appropriate course-of-action decisions but also decision-making to be made at the appropriate levels. Since each command rates and scores its own assets, these AC metrics can be used within a command to determine proper course of actions for command-level event responses. This can further ease the burden of the cyber warrior that needs to monitor millions of data points.

Future work will be required to incorporate asset state into the environment and to determine proper course-of-action using all this information. Knowing the state of the assets would provide guidance on how to respond to the attack properly. For instance, if the IA posture of the weapons system is secure (i.e. has been patched against vulnerability that the attack is exploiting), then the proper response may be to do nothing. If the generic desktop's state is not as secure, a different response may be required, despite both assets undergoing the same type of attack.

References

[1] Mark Kagan, "Cyberdefenders Protect Navy Networks," *Military Information Technology (MIT)*, vol. 13, no. 11, December 2009.

[2] Maryann Lawlor, "Defeating Sophisticated Threats Requires Multipronged Tactics," *AFCEA International Signal Online*, pp. http://www.afcea.org/signal/articles/templates/Signal_Article_Template.asp?articleid=1307&zoneid=207, 2007.

[3] U.S. Department of Defense. [Online]. http://www.defense.gov/about/dod101.aspx

[4] DoD CIO, "Vision for a Net-Centric, Service-Oriented DoD Enterprise (v1.0)," Department of Defense, Washington, DC, 2007.

[5] The White House. [Online]. http://www.whitehouse.gov/about/air-force-one

[6] Wikipedia. Triage. [Online]. http://en.wikipedia.org/wiki/Triage

[7] M.D., Richard Ellenbogen, M.D., Christopher I. Shaffrey, M.D. Diana Barrett Wiseman, "Triage for the NeuroSurgeon: Military Triage," *Neurosurg Focus*, vol. 12, no. 3, 2002.

[8] John R. Goodall, Anita D'Amico, and Jason K. Kopylec, "CAMUS: Automatically Mapping Cyber Assets to Missions and Users," in *Military Communications Conference (MILCOM)*, Boston, 2009.

[9] Luc Beaudoin and P. Eng, "Asset Valuation Technique for Network Management and Security," in *6th Intl. Conference on Data Mining - Workshops*, 2006, pp. 718-721.

[10] Reginald E. Sawilla and Xinming Ou, "Identifying Critical Attack Assets in Dependency Attack Graphs," in *Proceedings of the 13th European Symposium on Research in Computer Security*, Malaga, Spain, 2008, pp. 18-34.

[11] David Austin, "How Google Finds Your Needle in the Web's Haystack," *American Mathematical Society*, pp. http://www.ams.org/featurecolumn/archive/pagerank.html, 2011. [Online]. http://www.ams.org/featurecolumn/archive/pagerank.html

[12] K. Paul, Ching-Lai Hwang Yoon, "Multiple Attribute Decision Making: an Introduction," *Sage University Paper series on Quantitative Applications in the Social Sciences*, pp. 07-104, 1995.

[13] Robert M. Bell, l Jim Bennett, Yehuda Koren, and Chris Volinsky, "The Million Dollar Programming Prize," *IEEE Spectrum*, May 2009.

[14] Gediminas Adomavicius and Alexander Tuzhilin, "Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, June 2005.

[15] Wikipedia. Factor Analysis. [Online]. http://en.wikipedia.org/wiki/Factor_analysis

[16] Wikipedia. Artificial Neural Networks. [Online]. http://en.wikipedia.org/wiki/Artificial_neural_network

[17] Gary Stonebumer, Alice Goguen, and Alexis Feringa, "Risk Management Guide for Information Technology Systems," Information Technology Lab, National Institute of Standards and Technology, Gaithersburg, MD, NIST Special Publication 800-30 2002.

[18] Charles P. Pfleeger and Shari Lawrence Pfleeger, "Risk Analysis," in *Security in Computing*. New Jersey: Prentice Hall, 2003.

[19] John Moteff, "Risk Management and Critical Infrastructure Protection: Assessing, Integrating, and Managing Threats, Vulnerabilities and Consequences," CRS Report for Congress 2005.

[20] Thomas L. Saaty, "Decision making with the analytical hierarchy process," *Intl. Journal of Services Sciences*, vol. 1, no. 1, pp. 83-98, 2008.

[21] Hsu-Shih Shih, Huan-Jyh Shyur, and Stanley E. Lee, "An extension of TOPSIS for Group Decision Making," *Mathematical and Computer Modeling*, vol. 45, no. 7-8, pp. 801-813, 2007.

[22] Jia-Wen Wang, Ching-Hsue Cheng, and Kun-Cheng Huang, "Fuzzy hierarchical TOPSIS for supplier selection," *Applied Soft Computing*, vol. 9, no. 1, pp. 377-386, January 2000.

[23] Wikipedia. Page Rank. [Online]. http://en.wikipedia.org/wiki/PageRank

[24] Mercedes Pascual Stefano Allesina, "Googling Food Webs: Can an Eigenvector Measure Species' Importance for Coextinctions?," *PLoS Computational Biology*, vol. 5, no. 9, September 2009.

[25] S. P. Borgatti M. G. Everett, "The Centrality of Groups and Classes," *Journal of Mathematical Sociology*, vol. 23, no. 3, pp. 181-201, 1999.

[26] Iacopo Carreras, Daniele Miorandi, Geoffrey S. Canright, and Kenth Engo-Monsen, "Eigenvector Centrality in Highly Partitioned Mobile Networks: Principles and Applications," *Studies in Computational Intelligence*, vol. 69, pp. 123-145, 2007.

[27] Irad Ben-Gal, "Bayesian Networks," in *Encyclopedia of Statistics in Quality and Reliability*. New York: Wiley and Sons, 2008.

[28] Janos Fülöp. Introduction to Decision Making Methods. [Online]. http://academic.evergreen.edu/projects/bdei/documents/decisionmakingmethods.pdf

[29] Mei-Tai Chu, Joseph Shyu, Gwo-Hshiung Tzeng, and Rajiv Khosla, "Comparison among three analytical methods for knowledge communities group-decision analysis," *Expert Systems with Applications*, vol. 33, pp. 1011-1024, 2007.

[30] R. Venkata Rao, *Decision Making in the Manufacturing Environment*. London: Springer-Verlag, 2007.

[31] Thomas L. Saaty, *The Analytical Hierarchy Process: planning, priority setting, resource allocation*. New York: McGraw-Hill, 1980.

[32] Evangelos Triantaphyllou, Multi-Criteria Decsion Making Methods: A Comparative Study, November 2000.

[33] Subrata Chakraborty and Chung-Hsing Yeh, "A simulation based comparitive study of normalization procedures in multiattribute decision making," in *6th WSEAS Intl. Conference on Artificial Intelligence, Knowledge Engineering and Databases*, Corfu Island, Greece, 2007.

[34] Kyung Sam Park Byeong Seok Ahn, "Comparing methods for multiattribute decision making with ordinal weights," *Computers and Operations Research*, vol. 35, no. 5, pp. 1660-1670, 2008.

[35] Thomas L. Saaty, "Fundamentals of the Analytic Network Process - Dependence and Feedback in Decision-Making with a Single Network," *Journal of Systems Science and Systems Engineering*, vol. 13, no. 2, pp. 129-157, June 2004.

[36] Peter Mell, Karen Scarfone, and Sasha Romanosky, "A Complete Guide to the Common Vulnerability Scoring System v2.0," 2007.

[37] Walt Heimerdinger, "Scyllarus Intrusion Detection Report Correlator and Analyzer," in *DARPA Information Survivability Conference and Exposition*, 2003, pp. 24-26.